**Performance: Perception or Reality?**

**CLARiiON Environment**

Octavio Palomino (Octavio@octaviopalomino.net)

EMC Global Services – Implementation Specialist

Implementation Engineer, CLARiiON Solutions Specialist (EMCIE)

EMC Solutions Architect –

2014 – Independent Technology Advisor

Javier Alvarez Dasnoy (AlvarezDasnoy\_Javier@emc.com)

 EMC Global Services – Implementation Delivery Engineer

Implementation Engineer, Storage Management Specialist (EMCIE)

**Table of Contents**

[Objectives 4](#_Toc271204072)

[Which Components should you Analyze to Address a Performance Issue? 4](#_Toc271204073)

[How to Work on a Performance Requirement from the Storage Area 5](#_Toc271204074)

[What do we Hear when the Customer Speaks about Performance? 8](#_Toc271204075)

[What do we Say when we Speak about Performance? 9](#_Toc271204076)

[Implementing a CLARiiON® Storage System with the Customer 12](#_Toc271204077)

[Which Data we Ask for and How we Interpret it 12](#_Toc271204078)

[What Needs to be Considered in the Implementation Phase and What is the Scope? 13](#_Toc271204079)

[Tips & Tricks for Performance Problems 15](#_Toc271204080)

[1-Collection of environmental data 16](#_Toc271204081)

[2-Collection of hardware-related data 16](#_Toc271204082)

[Conclusion 18](#_Toc271204083)

[ANEXO 1 – Performance Manager metrics for CLARiiON 19](#_Toc271204084)

[Documentation 22](#_Toc271204085)

*Disclaimer: The views, processes, or methodologies published in this compilation are those of the authors.  They do not necessarily reflect EMC Corporation’s views, processes, or methodologies.*

# Objectives

Identifying a performance-related issue begins with human perception; a customer calls and says, “Our systems are slow; we believe there is a performance problem in the storage system.”

However, as EMC Solutions Architect José Brunfman told us, “It doesn’t matter if there is a real problem or not. From the moment the customer complains, there is a problem, whether it comes from the customer’s imagination or from an erroneous interpretation of reality.”

In this article, we will present an approach to translate “I feel that the system is slow” into quantifiable and comparable data obtained objectively.

# Which Components should you Analyze to Address a Performance Issue?

 After a customer informs you that “the system is slow,” you should investigate to determine “slow as compared to what”?

Sometimes ‘slowness’ is just a perception, but sometimes it is a real problem caused between the final user and the beginning of the requirement. Here is a flow diagram of the process:

Beginning of requirement 🡪 Application 🡪 Operating System (OS) 🡪 File System Software 🡪 Multipathing Software 🡪 Host Bus Adapter (HBA) 🡪 Fiber Channel (FC) 🡪 FC Switch 🡪 FC 🡪 Storage Front-end Adapter (FA) 🡪 Storage Cache 🡪 Storage Back-End (BE) -> Physical Disks (usually with RAID protection) 🡪 BE 🡪 Storage Cache 🡪 FA 🡪 FC 🡪 Switch 🡪 FC 🡪 HBA 🡪 Multipathing Software 🡪 File System Software 🡪 OS 🡪 Application 🡪 Requirement’s response.

As you can see, there are many components that can generate a delay in a requirement. A serious analysis requires the involvement and commitment of many areas of the company.

These areas usually include Storage Administration, OS Administration, Database (DB) Administration, Application Development, and Network / Storage Area Network (SAN) Administration.

The end user does not investigate the issue, but provides input about the “system slowness” and is the recipient of analysis results.

# How to Work on a Performance Requirement from the Storage Area

This article focuses on the procedures, tips, and tricks you can use when analyzing a performance problem from the storage point of view. EMC ControlCenter® Performance Manager is the tool used to gather information about the environment.

A baseline is the core of this analysis. It is a basic standard that allows us to compare resource behavior over time.

Here are some considerations when creating a baseline using EMC ControlCenter Performance Manager as the tool to gather data.

1 – Physical Components

**Metrics tab**

* From the LunStats metric, choose the N LUNs with more total I/Os per second. N should be a value that comprises approximately 80% of the system I/Os. If there are too many devices, choose only those that represent a significant percentage of the total I/O.



Consider the Read Size (KB) / Write Size (KB), Read Cache Miss, Write Cache Miss, and Cache Hits. This information helps us to understand the application’s behavior and how the application is taking advantage of cache. Response time is another important metric (the amount of time that the array is using to answer a requirement).

* From the StorageProcessorStats – Consider metrics such as % Utilization since this metric contains all ports of every SP.



2 – Timeframe considerations

As a general guideline, utilize the weekly btp files to create a baseline. However, if there are relevant business processes or I/O peaks that occur in different timeframes (for example, at the end of the month or quarter), it is important to consider and analyze them separately.

There are two approaches for this type of I/O. Analyze a unique timeframe that includes normal and peak periods, or analyze the peak and normal periods separately. The problem with the first approach is that you will lose visibility of the peak periods, as they will be averaged with the normal periods.

You can also use the WLA Analyst policy in ControlCenter to create specific btp files for performance issues or business processes that occur in a specific time period. This allows you to have a snapshot of all the components in that specific period of time.

3 – Timeframe considerations

Save the btp files in a separate location before they are deleted by the ControlCenter WLA Retention Policy.

4 – Host perspective

Finally, include host information in the baseline, whether using information from Performance Manager from a host with an EMC ControlCenter agent or using host-based tools as Performance Monitor.

With this information, you will see the normal behavior of the installation. This should be a storage administrator’s routine task. Initiating a baseline analysis when the problem already exists is useless.

Do not work with too many variables, as the purpose of this article is to define a process to obtain and analyze the information and to select the best to understand the performance issue. We don’t intend to perform a deep performance analysis.

# What do we Hear when the Customer Speaks about Performance?

The customer’s message is in a particular context. We need to situate that context and obtain the greatest amount of information possible regarding changes that have occurred beyond our function. These may include server hardware changes, application updates, software upgrades, or changes in business processes.

There is no unique way to obtain this information, but if the company has adopted Information Technology Infrastructure Library (ITIL) guidelines or a similar framework, there will be a change document (for example, a Request for Change (RFC) document).

During our conversation with the customer, it is important to identify the frequency of the problem, if it arises in a specific timeframe, and which platforms are affected (databases, batch processes, etc).

We must interpret what the customer is saying, and interpret what the data from the hardware equipment is showing.

If we listen to the customer, we will have valuable information about the context when reading and interpreting objective data. For example, if a full backup is in progress, we will see very high read values that do not correspond to normal behavior. If we didn’t know this information, we could erroneously think that there was a problem.

# What do we Say when we Speak about Performance?

We start our analysis after listening to the customer and gathering information. Separate the components that are important for a performance issue and analyze their behavior as if they were isolated components (we will consolidate the information later).

 The components that we need to analyze are:

1. HBA Configuration Parameters
2. Cache Size. R/W Hits/Misses
3. Protection types used: RAID 1/0, RAID 5, RAID 6, etc.
4. R/W Block Sizes
5. Number of I/Ops – R/W
6. Hosts – Number of I/Ops – R/W
7. Hosts – Memory
8. Hosts – Processor usage

To begin, we need general information regarding the affected hosts using EMCReport (for Windows) or EMCGrab (for Unix-like OS). With this information, we will check the support matrix for interoperability and for HBA parameters using Host Environment Analysis Tools (HEAT) (point 1).

Regarding points 6, 7, and 8, we will only be able to get this information if there is an EMC ControlCenter agent installed on the host. Another option is to ask the host administrator for the results of Performance Monitor (for Windows) or sar (for Unix).

Points 2, 3, 4, and 5 refer to the Storage System, and should be analyzed using EMC ControlCenter Performance Manager. The CLARiiON® agent license is required and at least the WLA Daily Data Collection Policy (DCP) enabled for this array. You can also obtain the data using Navisphere Analyzer, but this is beyond the scope of this article.

If the problem is occurring at the moment the analysis is being performed, enable the WLA Revolving Data DCP and after 30 minutes obtain the data using the option “get revolving data” in ControlCenter. This will generate a file containing the data for that specific time frame.

We will start discussing the performance problem after gathering all the information. The terms used in this discussion are:

* **Alignment –** Data block addresses compared to RAID stripe addresses
* **Coalesce –** To combine multiple smaller I/O into one larger I/O
* **Concurrency –** More than one application or thread writing to a LUN or disk at the same time
* **Flush –** Data in write cache written to disk
* **Locality –** Multiple I/O requested from a reasonably small area on the disk (same MB or GB)
* **RDBMS –** Relational Database Management System
* **“Throughput”—**IOPS: typically important for filesystem access, RDBMS; small requests (2-16KB)
* **“Bandwidth”—**MB/s: typically important for backups, DSS operations, rich media access (64KB, 256KB)
* **“Response time”—**a key measurement of quality of service (QoS); an array can offer a high max IOPS figure, but deliver consistently slow response time
* **Prefetch –** The act of reading data into read cache before the host requests it. Normally only found with sequential accesses
* **Request size –** The I/O size of a read or write requested by a host
* **Burstiness –** A measure of the I/O pattern, characterized by brief periods of intense activity followed by periods that are less busy
* **Write-aside –** The act of bypassing write cache and writing directly to physical disk; found with writes larger than a preconfigured size
* **Read cache hit –** Data is found in read cache or write cache
* **Read cache miss –** Data must be fetched from disk
* **Write cache hit –** Data is written into an available empty page in cache (ONLY)
* **Write cache miss –** Data written to disk (write-aside, or cache disabled)
	+ Data causes a forced flush (no page is available until flushed)
* **Spindles –** Every physical disk has a spindle. Spindle speed is a limiting factor according to the following tables:





Storage system performance must be planned at the implementation phase.

# Implementing a CLARiiON Storage System with the Customer

There was an implementation before any performance problem. At this point, we will discuss considerations when implementing a new CLARiiON Storage System.

Initially, we have to identify the applications running on the array:

* Will it receive an existing operating environment? That is, will an application that is currently running on the company will be migrated to run on the new storage system?
* Will it be a completely new environment? That is, will a new application be developed based on business requirements?
* Is it possible to compare, collect data, and predict future behavior? We need to get information that allows us to predict the behavior of what will be implemented.

# Which Data we Ask for and How we Interpret it

This is the information we need to gather:

1. I/Ops from applications. Identified in Read/Writes.
2. Application types: For example, which type of database (Oracle, SQL Server, Universe, etc.)? Each type of database has different behaviors and data structures. For example, SQL Server has three structures: DBF, tempdb, and transaction logs. Oracle has DBF, Redo logs, Control Files, indexes, etc.
3. Behavior: Oriented to write, read, random, and/or sequential
4. Growth rate: Consider the growth rate of the application when creating Metaluns. A metalun cannot be extended in a RAID Group which already has LUNs that form that metalun. That is, if we have a 200 GB metalun with 100 GB on RAID Group 1 and 100 GB on RAID Group 2, we will not be able to extend that metalun on these RAID Groups because a performance problem will be created (see FLARE Knowledgebook).
5. Operating Systems: In the case of Windows, verify block size to avoid alignment issues.
6. Disks: Types of disks and number and speed of spindles.

After gathering this information, we will create tables to calculate the necessary spindles and predict CLARiiON behavior.

The implementation phase is the basis for the behavior of the CLARiiON in its lifespan. Our implementation tasks and the customer commitment to maintain, gather data, and remain aligned to the best practices cited in this article will guarantee a positive overall experience and satisfaction with the storage system.

Planning for performance is a dynamic process. The solution implemented with a new storage system does not necessarily correspond with future business demands. These changing demands must be identified by collecting new information, actualizing baselines, and interpreting data.

CLARiiON has tools that allow us to modify the behavior of the array through its lifespan, for example, using LUN Migrate or expanding Metaluns (see 1024\_emc\_clariion\_metaluns\_ldb.pdf)

Use Powerlink documentation about products and implementation environments’ recommendations (databases, operating systems), knowledge books (MirrorView™, SnapView™, etc) and FLARE® versions.

# What Needs to be Considered in the Implementation Phase and What is the Scope?

At the moment of implementation, we have the information that allows us to create a table with the following information:



The column IOPS worst case (wc) records the total I/O, calculated in the following way:

Our example is the MAX\_1 row, one of the four maximum values that were observed.

MAX\_1 IOPS is 716 is obtained from the sum of the read and write columns (231+485). As the protection is RAID 5, the final value will be:

 IOPS Worst Case (WC) = iops Read + 4 \* iops write

 IOPS Worst Case (WC) = 231 + 4 \* 485 = 2171

We will not consider other variables in this analysis (other than iops and RAID). It is very difficult to get data about cache usage or HBA configuration parameters in a new implementation. However, if this data is available, consider it in the analysis.

Nevertheless, the storage cache is configured at the storage system level and not at the LUN level. It is not advisable to modify the recommended settings (1/3 for read cache and 2/3 for write cache).

If the storage system is already in a production environment when the problem arises, consider these points:

* Predominant I/O request sizes
	+ Small Up to 16KB, Large > 64 KB
* Read/write ratio
	+ Writes consumes more resources than reads
* *Host* load to *disk* load based on your RAID type
	+ RAID 5: Total I/O = Host Reads + 4 \* Host Writes
	+ RAID 6: Total I/O = Host Reads + 6 \* Host Writes
	+ RAID 1 and RAID 1/0: Total I/O = Host Reads + 2 \* Host Writes
* Random (non-contiguous R/W), sequential access or mixed
	+ Random uses more storage resources than sequential
	+ Response time
	+ Prefetch
* Cache
	+ Read Cache Miss / Hits
	+ Write Cache Miss / Hits
* HBA Configuration as outstanding

# Tips & Tricks for Performance Problems

**(or how to proceed when there is a performance related request)**

When we have a performance request, we need to initiate the analysis on our equipment as if the problem existed. The idea is to have a unified approach and look for something, instead of gathering data with no specific guideline in mind. The best that can happen after finishing the analysis is to demonstrate that there wasn’t a problem.

The steps for this analysis should be:

**Baseline:** The baseline is essential to identify an abnormal behavior in the timeframe that the customer states there is a problem. It is important to understand that no single graph will show a problem, only signs of abnormal behavior. In addition, an increase in the utilization of some components of the array can be caused by a modification of the reality (for example, an application can have more I/O because the business is growing). If the increase in I/O is significant, it can cause a performance problem that needs to be addressed.

**Data to Consider:** Contrast the Vital Signs between the baseline and the current data to look for significant differences. The same data sources and timeframes should be used when doing the comparison. We will analyze the new data from this point.

The first step is to correlate the host and the storage system.

Let’s present this approach using an example: suppose that we have a performance problem in the database EMCBASE.

## 1-Collection of environmental data

* Host: This database runs on the host emchost
* Storage: It resides on a CX4, using ports SPA0 and SPB0. These ports are shared with host EMCBACKUP
* Switch: We need to identify the switch ports to which the host is connected. In this case, we have 4 ports: 2 for the host and 2 for the storage.

## 2-Collection of hardware-related data

 For each component (host, switch, storage) we use different types of data.

* Host perspective: It is important to analyze the response time from the host perspective. Although this information alone is not enough to form a conclusion, it allows us to verify if the user’s perception corresponds with the host data. Nevertheless, a high value of response time or wait does not necessarily mean that there is a problem in a disk (for example, in a recent issue, we had a DB with very high response time values. The cause of the problem was identified as an ntp process).
* Switch perspective: We analyze the utilization percentage of the ports previously identified to see if we are approaching the saturation point. We can also look for port errors for signal loss, synch loss, or frame loss that could signal physical problems.
* Storage System: We need to analyze the ports and the set of disks associated to the BD. Compare the current results to the baseline data.
	+ **Ports:** Consider if the port is shared between other applications. In that case, analyze which volume of I/O this application is adding to the port. It is important to analyze the usage percentage to determine if we exceed the 80% threshold.
	+ **Devices:** I/Os per second is the first variable weneed to compare with the baseline.

There are also many variables that are of interest:

* + - Samples: this information is partial and subjective, but allows us to identify abnormal values compared to the baseline.
		- Misses: Misses mean that the application is accessing the disks, and we know that accessing the disks is slower than accessing cache. However, this doesn’t necessarily mean that it is a problem. Maybe the application behaves that way over time, which is what we need to verify when comparing to our baseline.
		- HA Kbytes: This variable shows the Host Bus Adapter throughput. The value should be close to the switch port throughput connected to that HBA (remember that you should have at least 2 paths connected to the same device).
		- Average Size: This variable shows the average size of the read and writes. Correlate this value to the fixed value of every specific vendor file system and the vendor database. In this way, you are able to identify the behavior of these applications (default value for CLARiiON is 8 K; we do not advise modifying this value in the array) .

# Conclusion

A performance analysis is a perception or a result. It is a perception because it is initiated by a customer action when experiencing an unusual delay, or a result when a specification (for example, a hardware failure) causes abnormal behavior until the problem is resolved. In this analysis, we initially considered the perception point of view. Based on feedback from the customer, we develop a method of analysis to verify if that perception was based on a real performance issue. Sometimes the results of a performance analysis do not match the customer’s expectations (maybe the customer expected a solution and we didn’t identify any problem). However, we can guarantee that we worked with all the available information. There is information that simply does not exist because performance issues cannot be analyzed linearly since many variables are involved.

We hope that this article unifies the analysis of performance issues. Our objective is to offer enough information to perform the initial analysis of a performance problem, and to perform this analysis objectively. The methodology is not a solution for every performance issue, but rather a guideline to prepare a document that can be shared with others in the company to identify possible root causes of performance issues or perceptions. The performance analysis must be performed collaboratively with all the company’s technology areas.

Finally, this article is only a basic guideline to analyze performance issues, it is not intended for low-level analysis. We wrote this article based on many years of experience in the field. If, after following the steps detailed in this document, you are not satisfied with the results, we advise escalating the problem to EMC. Specific groups of performance specialists and gurus have more experience and tools to analyze your issue.

The examples in this article were performed by EMC field personnel (IDE and IS), and it is our humble wish that it can be useful for anyone taking their first steps in Storage Administration and Implementation.

# ANEXO 1 – Performance Manager metrics for CLARiiON

**CLARiiON disk metrics**

|  |  |
| --- | --- |
| ios per sec | reads per sec + writes per sec |
| reads per sec | Total number of read requests processed by this disk per second. |
| writes per sec | Total number of write requests processed by this disk. |

**CLARiiON LUN metrics**

|  |  |
| --- | --- |
| % hits |  100 \* (hits per sec / host ios per sec) |
| % read hits | 100 \* (Read cache hits per sec /host read requests per sec) |
| % write hits | 100 \* (Write cache hits per sec /host write requests per sec) |
| blocks per disk read | disk blocks read per sec / disk reads per sec |
| blocks per disk write | host blocks written per sec / disk writes per sec |
| blocks per host read | host blocks read per sec / host read requests per sec |
| blocks per host write | host blocks written per sec / host write requests per sec |
| blocks per write | host blocks written per sec / host write requests per sec |
| Blocks Prefetched per sec | The number of blocks that were prefetched by the read cache. |
| Busy Ticks per sec | The number of ticks that this LUN was busy. |
| disk blocks per sec | disk blocks read per sec + disk blocks written per sec |
| disk blocks read per sec | Disk blocks (512 bytes) read from LUN per second. |
| disk blocks written per sec | Disk blocks (512 bytes) written to LUN per second. |
| disk ios per sec | disk reads per sec + disk writes per sec |
| disk reads per sec | Number of reads per second. |
| disk writes per sec | Number of writes per second. |
| Fast Writes per sec | The number of writes that didn't access disk (write to empty page of write cache or overwrite existing data). |
| Forced Flushes per sec | The number of times a write had to flush one or more pages to make room in the cache. This counts as a cache miss.  |
| hits per sec | read cache hits per sec + write cache hits per sec |
| host blocks per sec | host blocks read per sec + host blocks written per sec |
| host blocks read per sec | Total number of disk blocks read by host from this LUN. |
| host blocks written per sec | Total # of disk blocks written by host to this LUN. |
| host ios per sec | host read requests per sec + host write requests per sec |
| host read requests per sec | Total number of read requests made by host to this LUN. |
| host write requests per sec | Total number of write requests made by host to this LUN. |
| Idle Ticks per sec | The number of ticks that this LUN was idle. |
| read cache hits per sec | The number of times a read was satisfied from the read or write cache. |
| Read 512 per sec | Number of single-sector reads per sec. One sector is equal to 512 bytes. |
| Read Less Than 2K per sec | The number of reads per second that were 2 and 3 sectors. |
| Read Less Than 4K per sec | The number of reads per second that were 4 through 7 sectors. |
| Read Less Than 8K per sec | The number of reads per second that were 8 through 15 sectors. |
| Read Less Than 16K per sec | The number of reads per second that were 16 through 31 sectors. |
| Read Less Than 32K per sec | The number of reads per second that were 32 through 63 sectors. |
| Read Less Than 64K per sec | The number of reads per second that were 64 through 127 sectors. |
| Read Less Than 128K per sec | The number of reads per second that were 128 through 255 sectors. |
| Read Less Than 256K per sec | The number of reads per second that were 256 through 511 sectors. |
| Read Less Than 512K per sec | The number of reads per second that were 512 through 1023 sectors. |
| Read Overflow per sec | The number of reads that were larger than 1,023 blocks. |
| Sum Queue Lengths On Arrival High | High-order 32 bits of queue length total value. |
| Sum Queue Lengths On Arrival Low | Low-order 32 bits of queue length total value. |
| write cache hits per sec | The number of write requests per second that were satisfied by the write cache since they have been referenced before and not yet flushed to the disks. Write cache hits occur when recently accessed data is referenced again while it is still in the write cache. |
| Write 512 per sec | Number of single-sector writes per second. One sector is equal to 512 bytes. |
| Write Less Than 2K per sec | The number of writes per second that were 2 and 3 sectors. |
| Write Less Than 4K per sec | The number of writes per second that were 4 through 7 sectors. |
| **CLARiiON LUN metrics** |
| Write Less Than 8K per sec | The number of writes per second that were 8 through 15 sectors. |
| Write Less Than 16K per sec | The number of writes per second that were 16 through 31 sectors. |
| Write Less Than 32K per sec | The number of writes per second that were 32 through 63 sectors. |
| Write Less Than 64K per sec | The number of writes per second that were 64 through 127 sectors. |
| Write Less Than 128K per sec | The number of writes per second that were 128 through 255 sectors. |
| Write Less Than 256K per sec | The number of writes per second that were 256 through 511 sectors. |
| Write Less Than 512K per sec | The number of writes per second that were 512 through 1023 sectors. |
| Write Overflow per sec | The number of writes that are larger than 512 blocks. |

**CLARiiON SP metrics**

|  |  |
| --- | --- |
| Arrivals To Non Zero | The number of times a user request arrived while at least one other request was being processed.  |
| blocks per read | FE Blocks read per sec / FE Reads per sec |
| blocks per write | FE Blocks Written per sec / FE Writes per sec |
| FE Blocks Read per sec | Disk blocks (512 bytes) read by SP per second. |
| FE Blocks Written per sec | Disk blocks (512 bytes) written by SP per second.  |
| FE ios per sec | FE Reads per sec + FE Writes per sec |
| FE Reads per sec | Total number of read requests processed by this SP per second. |
| FE Writes per sec | Total number of write requests processed by this SP per second. |
| Percent Dirty Pages | Percentage of pages owned by this SP that were modified since they were last written to disk. |
| SP Busy Ticks per sec | The number of clock interrupts on the SP board that occurred with the SP not in its idle state. 1tick = 100 ms. |
| SP Idle Ticks per sec | The number of clock interrupts on the SP board which have occurred with the SP in its idle state. 1 tick = 100ms, |

# Documentation

* CLARiiON Top 10 Best Practice Hints for Performance
* H2358\_clariion\_best\_prac\_fibre\_chnl\_wp\_ldv
* H1049\_emc\_clariion\_fibre\_channel\_storage\_fundamentals\_ldv
* EMCClariion\_SystemConfigurationGuide.pdf
* H1024\_emc\_clariion\_metaluns\_ldv
* EMC Clariion Best Practices for Performance and Availability
* H4208-an-intro-emc-clariion-hard0drive-tech-wp
* h6099-navisphere-performance-in-large-configurations-wp